DOCUMENT RESUME

ED 320 439 FL 018 603

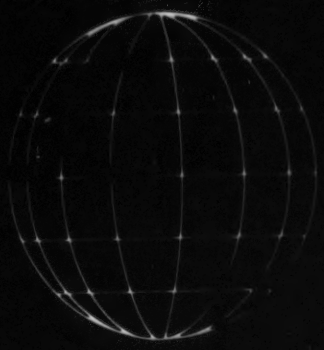
AUTHOR Ginsberg, Ralph B.
TITLE Implications for Intelligent Tutoring Systems for Research and Practice in Foreign Language Learning, NFLC Occasional Papers.
INSTITUTION Johns Hopkins Univ., Washington, DC. National Foreign Language Center.
PUB DATE Mar 89
NOTE 22p.; Revised version of a paper presented at a Conference on Empirical Research on Second Language Learning in Instructional Settings (Bellagio, Italy, June 20-24, 1988).
AVAILABLE FROM Cashier, National Foreign Language Center, 1619 Massachusetts Ave., N.W., Washington, DC 20036 ($3.00 including postage and handling; checks payable to NFLC).
PUB TYPE Information Analyses (070) -- Reports - Evaluative/Feasibility (142) -- Speeches/Conference Papers (150)

EDRS PRICE MF01/PC01 Plus Postage.
DESCRIPTORS *Artificial Intelligence; *Computer Assisted Instruction; *Educational Environment; *Educational Research; Instructional Effectiveness; Language Research; Second Language Instruction; Second Language Learning

ABSTRACT

Most of the now commonplace computer-assisted instruction (CAI) uses computers to increase the capacity to perform logical, numerical, and symbolic computations. However, computers are an interactive and potentially intelligent medium. The implications of artificial intelligence (AI) for learning are more radical than those for traditional CAI. AI research and development promises to deepen understanding and reshape practice in and out of the classroom. This report, the first of three, gives an overview of the character and motivation of AI approaches to education and learning. The first section describes and gives the background for the report. The second section introduces AI programs designed to emulate good teaching (intelligent tutoring systems, or ITS), and computer-based "microworlds," designed to be effective learning environments. Both are compared with the more familiar CAI and simulation. In section three, the main components of ITS architecture are discussed, including domain expertise, student modeling and diagnosis, teaching strategies, planning and control, task structure, and the learner-computer interface. Section four is concerned with how learning has been conceptualized by AI researchers, and with the design principles that derive from that conceptualization. Finally, issues concerning research and industrial design are outlined. (MSE)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****



"PERMISSION TO REPRODUCE THIS MATERIAL HAS BEEN GRANTED BY
R. Lambert
TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)."

U.S. DEPARTMENT OF EDUCATION
Office of Educational Research and Improvement
EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)
X This document has been reproduced as received from the person or organization originating it.
☐ Minor changes have been made to improve reproduction quality.
• Points of view or opinions stated in this document do not necessarily represent official OERI position or policy.

IMPLICATIONS OF INTELLIGENT TUTORING SYSTEMS FOR RESEARCH AND PRACTICE IN FOREIGN LANGUAGE LEARNING

Ralph B. Ginsberg
National Foreign Language Center

COMPLIMENTARY COPY

March 1989

FL018603

This paper is a revised version of a paper presented at the Conference on Empirical Research on Second Language Learning in Instructional Settings held at the Rockefeller Study and Conference Center, Bellagio, Italy, June 20-24, 1988, and sponsored by NFLC and the European Cultural Foundation. It is the first in a series of three papers on artificial intelligence and its implications for designing and analyzing foreign language learning environments and teaching systems.

About the Author

Ralph B. Ginsberg, Deputy Director of the National Foreign Language Center, is Professor of Regional Science and Chairman of the Regional Science Department at the University of Pennsylvania where he was also formerly Graduate Group Chairman in Public Policy Analysis. Dr. Ginsberg directs the NFLC's research program in applications of advanced, computer-based technologies to foreign language learning. He has worked extensively in applied probability and data analysis, and in organization theory, and is the author of several books and articles in these areas. He has been Editor of the *Journal of Mathematical Sociology* and serves on the editorial boards of several international journals. He has been Visiting Professor at Princeton University, Erasmus University (Netherlands), and Umea University (Sweden). Dr. Ginsberg received his Ph.D. in Sociology from Columbia University.

About the *Occasional Papers*

This is the third in a series of *Occasional Papers* published by the National Foreign Language Center (NFLC). Further papers are available for $3.00 each, including postage and handling, or $18.00 for six issues. Papers may be obtained by sending a check and the name of the paper desired to Cashier, National Foreign Language Center, 1619 Massachusetts Ave. NW, Washington, DC 20036. Checks should be made payable to NFLC.

NFLC prints and distributes as *Occasional Papers* articles on a wide variety of topics related to foreign language research, education, and policy. The papers are read by an internal review committee. The *Occasional Papers* are intended to serve as a vehicle of communication and stimulant for discussion among groups concerned with these topic areas. The views expressed in these papers are the responsibility of the authors and do not necessarily reflect the views of NFLC or of The Johns Hopkins University.

Copyright 1989, The National Foreign Language Center
at The Johns Hopkins University

Implications of Intelligent Tutoring Systems for Research and Practice in Foreign Language Learning

RALPH B. GINSBERG

National Foreign Language Center

1. INTRODUCTION

In this and two subsequent papers I shall explore some implications of artificial intelligence for the design and empirical analysis of learning environments and teaching systems for foreign languages. At the same time, since artificial intelligence is implemented on computers or in settings in which computers play a fundamental role (such as hypermedia with interactive video and voice synthesis), I shall be discussing foreign language learning that takes place in designed environments that are for the most part neither classroom based nor classroom managed. The issues with which I shall be concerned here, however, do not not depend in any important way on the technology of computers or classrooms, and accordingly the research I shall review is pertinent to traditional teaching methods as well as computer-based learning.

Computers are, in one respect, powerful tools which vastly increase our capacity to perform logical, numerical, and symbolic computations. Most of the computer-aided instruction (CAI) that is now commonplace in virtually every area of education uses them in this way. But in another respect computers are an interactive and potentially intelligent medium within which we can carry out our most important social and cognitive activities. Artificial intelligence (AI) is the branch of computer science which tries to exploit this intelligence and interactiveness. With regard to the transmission and acquisition of knowledge, this entails addressing two basic questions:

- Learning environments: what are the characteristics of the physical and social settings in which people learn efficiently and effectively, and how can such settings be constructed or simulated?
- Knowledge communication: how is knowledge successfully communicated and skill successfully imparted, and how can those communication processes be emulated and enhanced?

In order to build computational models addressing these issues, AI researchers have had to look deeply into the nature of knowledge, learning, and teaching, in the light of what is now possible with new computer-based technologies. Moreover, building functional intelligent programs or operational interactive environments requires that issues in the design of learning environments and knowledge communication be addressed in considerable specificity and detail. In this second respect, as a cognitive science whose models are heavily influenced by theories of computation, AI's implications for learning are more radical than those of "traditional" CAI, going well beyond the provision of tools to support current educational practice (Papert, 1980; Brown and Greeno, 1984; Schank, 1984; Pea and Soloway, 1987; and, for a cautionary note, Winograd and Flores, 1985). Thus, quite apart from the ultimate success of its artifacts, the process of AI research and development holds the promise of deepening our understanding, and hence of reshaping practice, in and out of the classroom. Indeed the theory and analysis that lies behind attempts to build AI programs is already having a major indirect influence on education through the nascent field of cognitive science, where, along with linguistics and psychology, AI is arguably first among equals (Haugeland, 1984).

Drill-and-practice tutorial programs for foreign language learning go back to the earliest days of CAI (see, e.g., Suppes, 1981). The lively interest in CALICO and similar organizations points to continued growth and diversity. With the exponential decrease in cost of computing power, with the development of powerful information retrieval methods and optical disk storage, with computer-driven hypermedia systems including audio, CD-ROM, and voice recognition available or just

around the corner, and with the increasing accessibility of multilingual word-processing software, computer graphics, and desktop publishing running on multitask workstations, the prospects are good for a new generation of innovative and exciting developments. In contrast to "traditional" CAI and these other newly emerging technologies, foreign language learning has not figured in any significant way in applications of AI to education, the bulk of attention, for one reason or another, being focused on science, mathematics, programming, and engineering. In the last few years some interesting prototypes using AI techniques and technologies for foreign language learning have been developed—Project ATHENA at MIT (Murray et al., 1987; Morgenstern, 1986); the CALLE project at Xerox (Feuerman et al., 1987; Xerox, 1985); the LINGER programs of Yazdani, Barchan, and their colleagues at Exeter (Barchan, 1987; Barchan et al., 1986; Barchan and Wusterman, 1988; Yazdani, 1988); and Cerri and Breuker's (1981) DART in the PLATO system—and several other projects are currently being planned. But I think it would be fair to say that these programs do not address the main issues raised in this paper, especially those related to learning and intelligent tutoring. Nevertheless, I shall argue that applications of AI to foreign language learning are feasible, and that the potential impact of AI is quite as great as that of any of the new hardware technologies themselves, because, directly or indirectly, AI can enable us to use these technologies more effectively.

This paper aims to give the reader a general overview of the character and motivation of AI approaches to education and learning. Its emphasis is on specifying how the key issues have been framed, sketching the theoretical underpinnings and main conceptualizations used by AI researchers, and surveying the alternative ways in which the issues have been addressed in practice. Although at various points, especially in section 3, applications to foreign language learning are suggested, detailed development of the implications is left for the two succeeding papers, the one dealing with the principal research issues and directions for design of foreign language instruction in computer-based environments, and the other dealing with the methodology of empirical research on foreign language learning.

The paper is organized as follows. The next section, motivated by the two questions raised above, introduces AI programs designed to emulate good teaching, known as "intelligent tutoring systems" (ITS), and computer-based "microworlds," designed to be effective learning environments. To get a better idea of what is different about AI, both are compared with their more familiar cousins, CAI and simulation. In section 3 the main components of ITS architecture, which are a convenient way of organizing the issues, are discussed. These include domain expertise, student modelling and diagnosis, teaching strategies, planning and control, task structure, and the learner-computer interface. Section 4 is concerned with how learning has been conceptualized by AI researchers, and with the design principles that derive from that. I shall not try to make a summary judgment about whether AI will or will not revolutionize foreign language learning: that would be premature, and in any case it is a complex question that turns on many factors besides the merits of the tools. Rather, I conclude by flagging some of the main points concerning research and instructional design which will be taken up in detail in the sequel.

2. INTELLIGENT TUTORING SYSTEMS, LEARNING ENVIRONMENTS, AND CAI

2.1 Overview and Examples

Applications of artificial intelligence to education (broadly conceived to include the learning of both cognitive and procedural skills, in and out of schools) cover a very wide range of subject matters, goals, and instructional styles. Even a simple classification is difficult, not to speak of a useful definition of "intelligence." Perhaps the most celebrated and controversial application of AI to education, LOGO, does not have any evident "artificial intelligence" at all, although, as its designer, Seymour Papert, has stressed (1980, Ch. 7), it is pervaded by AI conceptualizations and tools. In this paper I shall focus on what have come to be known as "intelligent tutoring systems" (ITS)—I prefer the term to the more usual "intelligent computer-aided instruction" (ICAI) because of the often unwarranted pejorative connotations of the latter—i.e. computer-based learning which in some sense is designed to emulate good teaching. (I shall, however, discuss computer-based "microworlds," like LOGO, where there is no tutorial intervention, briefly in section 2.3.) ITSs operate in a middle ground between completely learner-directed, highly contextualized "natural" learning environments, on the one hand, and strongly teacher-directed, decontextualized classrooms, on the other. They draw their inspiration from and illuminate important characteristics of both of these extremes.

The aim of ITSs, as Anderson, Boyle, and Reiser (1985) have succinctly put it, is to "provide the student with the same instructional advantage that a sophisticated human tutor can provide." They are intelligent, to the extent that they are successful, in the sense that their behavior would be recognized as "good teaching," i.e. in the sense that they can emulate that complex, intelligent human behavior. ITS originated in the early 1970s in the work of Carbonell (1970) as an attempt to overcome some of the limitations and rigidities inherent in what was then, and still is, the dominant form of computer instruction (CAI), viz. a set of stored textual presentations, explanations, exercises, responses to student input, etc., often called "frames," presented to the student according to predefined branching rules which incorporate a definition of the curriculum, ideas about tutoring and

remediation, and anticipated student responses. Although the behavior of such programs as perceived by the student is often not that different from ITS (Lewis et al., 1987), ITS and CAI differ fundamentally in the way the knowledge they possess is represented and in the way the components of the program are put together. I shall return to the contrast between ITS and CAI in the next subsection.

Since the pioneering work of Carbonell a number of intelligent systems have been built. Their teaching methods include help facilities that try to figure out what the learner really wants to know, coaching, Socratic and case method dialogues, and highly structured and directive tutoring. Most have remained as prototypes, partly because of the great expense involved in building them (see Pea and Soloway, 1987, for estimates and comparisons with the cost of CAI; and Johnson, 1988, for pragmatic considerations limiting the building of programs), and partly because they were designed primarily for research purposes in the first place. It is the results of this research—the evolving formulation of the key issues to be addressed, the range of methods developed to resolve them, and the experience gained from successes and failures along the way—rather than a "bottom line" evaluation of how successful they have been in changing the educational outcomes in existing school and training settings, that are of primary interest for researchers and designers of foreign language instruction. There are, however, several programs currently in use, including:

- STEAMER (Hollan et al., 1984), which makes use of computer graphics and simulation to help students learn the operation of shipboard steam propulsion systems;
- SOPHIE (Burton et al., 1982), designed to teach various aspects of electronic troubleshooting;
- WEST (Burton and Brown, 1982) and WUSOR (Goldstein, 1982), coaches for the computer-based games WEST ("How the West Was Won") and WUMPUS;
- GUIDON (Clancey, 1987), designed to teach medical diagnosis through case method dialogues; and
- LISP and Geometry tutors (Anderson and Reiser, 1985; Anderson, Boyle, and Yost, 1985), for the AI computer language LISP and high school geometry.

Several other programs are currently in production.

Work in ITS has progressed at a steady pace. The state-of-the-art in 1982 was summarized in an important book edited by Sleeman and Brown, where most of the work to that date on both sides of the Atlantic was represented. In the past few years there has been an explosion of books and articles in the area: two thoughtful and comprehensive books reviewing the field (Wenger, 1987; and Polson and Richardson, 1988), three edited volumes (Kearsley, 1987; Psotka et al., 1988; Mandl and Lesgold, 1988), and several useful review articles (e.g. Anderson, Boyle, and Reiser, 1985; Olsson, 1986; Dede et al., 1985), with many more in press.

2.2 ITS and CAI

The goals and aspirations of ITS can, perhaps, best be grasped by contrasting it with the familiar CAI programs it is meant to improve. ITS's improvements move it closer to successful human teachers and supportive learning environments, in and out of the classroom, and in this respect they bear on general issues of research in foreign language learning.

One potentially important drawback of CAI—one should not exaggerate how important this or the other drawbacks noted below really are for any particular pedagogical goal or subject matter—is its rigidity. "Traditional" CAI (the quotation marks signal caricature of both CAI and ITS) can be thought of as a directed graph or flowchart, consisting of nodes representing textual, graphical, and audiovisual presentations of material, menus, questions with their answer categories, error messages and tutorial explanations, etc.; and a set of links connecting each node to the program's next actions (presentations). The links embody a detailed specification of the flow of control, as determined by students' responses at the originating node or by some other, prespecified branching mechanism. The course author, perhaps assisted by an authoring program, must explicitly enter all of the nodes and specify all of the links (student responses and tutorial reactions). A session then consists of one of the possible paths through the graph. Such a scheme is satisfactory if the student needs only drill and practice, exercises, and an occasional tutorial. But it puts a great burden on the designer to anticipate all contingencies, a burden that is difficult to bear for rich, nonmechanical activities, like using a foreign language. By contrast, like human teachers, ITS tries to plan sessions with the student on the fly (Peachey and MaCalla, 1986):

- CAI requires a detailed predetermined course graph structure, while ITS constructs it dynamically as the session progresses;
- the fixed CAI branching must be chosen with a wide range of students in mind, while ITS teaching plans and branching actions are tailored to each student;
- CAI must anticipate all relevant events, while an ITS can revise its plan and start from an incomplete specification;
- CAI's characterization of the student is one of degree of competency obtained and its analysis of student responses is based simply on answer matching, while ITS can interact with a much more subtle model of student.

ITS, then, tries to respond more flexibly than CAI. But it also tries to teach more complex skills, and this has important consequences for how it must be designed. For simple skills and highly structured, low-level tasks, responses can be predicted and paths constrained. But for more complex skills and more integrative tasks (e.g., mathematics problem solving, writing, or carrying on a conversation in a foreign language) students must play a

more active role in the process, experimenting with various aspects of the domain and determining their own courses of action. Such student behavior may well be impossible to anticipate in detail. Thus, if the tutor is to be helpful, it must, like human tutors, be able to solve problems as they arise, i.e., it must have its own knowledge of the subject matter and be able to put it to use. Moreover, with complex skills, inferences from what the learner does to what he knows (can do) and why he does it are not at all straightforward. Expert knowledge of the subject matter and more complex, sensitive representations of the student are a second major difference between ITS and CAI.

A third, more subtle difference has to do with the way knowledge of the subject matter is represented in ITS and CAI. In CAI the knowledge is contained in the procedures, i.e. the branching rules, which contain the possible answers to exercises and drills (including the correct one) as conditions and the CAI responses as actions. Moreover in the branching rules teaching knowledge and subject matter knowledge (condition and action) are tied strongly together. ITS programming structures follow a different strategy which has proved successful in other AI applications (e.g. expert systems, computer vision, and natural language processing). First, the teaching and domain knowledge are separated into different modules—so that each can be independently modified and they can be more flexibly combined as the planning mechanisms require. Second, within the teaching and domain modules, expert knowledge is often represented "declaratively," as a set of facts and rules in a knowledge base, which can be modified independently of procedures that use them; inference rules and other computational mechanisms are then provided to access the knowledge base and put it to use. It is from the representation of expert knowledge and the separation of teaching and subject matter knowledge that ITS derives its power, and indeed these features are what makes it possible.[1]

A final difference between ITS and CAI has to do with the kinds of things they try to teach and the instructional principles they use to do so. Largely through the influence of such ITS researchers as John Seely Brown, Richard Burton, and Allan Collins, and the seminal, closely related work of Seymour Papert, our views of what does and should go on in classrooms or on computers, and how the learning process should be organized, are being transformed. In particular (see Pea and Soloway, 1987) a fact-oriented, classroom-based, and classroom-managed "transmission view of knowledge," in which the major pedagogical activity is the presentation of well-structured material to be learned through lecture, demonstration, recitation, drill, and practice—the setting and ethos of CAI!—is giving way to a more learner-centered view, in which an active learner, using prior understandings ("frames," "schemas," and "mental models") and a variety of domain-specific and general strategies, acquires knowledge in contexts which are isomorphic to the situations in which that knowledge will eventually be used (Collins et al., 1987; Papert, 1980; and a vast, often polemical literature). This "situated" view of knowledge and learning, which draws its "success models" from such settings as apprenticeship, collaborative work, and games (IRL, 1987)—and which evokes debates about "learning versus acquisition" and "immersion versus drill and practice" in foreign language pedagogy—is at the heart of many ITS applications, and has clear implications for foreign language learning that will be elaborated in the sequel.[2]

2.3 Learning Environments and Microworlds

Although this paper is primarily concerned with ITS, it would be useful to digress somewhat at this point to consider another important line of applications of AI ideas and techniques to education, the design and construction of computer-based microworlds. Microworlds intersect ITS in the "environment" and "interface" components of ITS architecture, to be discussed in section 3.6. They are of interest for foreign language learning not only in their own right, as analogs of "authentic" materials and content-based learning, but also because of the insights into learning in "natural" settings that have been generated by their designers, especially Papert (1980).

It is hard to give a crisp definition of the concept of a "microworld." The meaning of the term lies more in the intentions and goals of the designers and in how microworlds are constructed than in what it describes. Pea's (1987) definition captures much of current usage: "A microworld is a structured environment that allows the learner to explore and manipulate a rule-governed universe, subject to specific assumptions and constraints, that serves as an analogical representation of some aspects of the natural world." The best known, but by no means the only, examples of computer-based microworlds are the LOGO programming language (Papert, 1980) and its extensions to "turtle" geometry (Abelson and diSessa, 1980) and more recently to Lego-LOGO (Papert, 1986). LOGO is now widely used in schools in the United States and the United Kingdom, and it has generated considerable theoretical and empirical research (see, e.g., Yazdani, 1984; O'Shea and Self, 1983; Lawler and Yazdani, 1987).

As mentioned above the role of artificial intelligence is not obvious as a child programs in LOGO. Nor is AI obvious in the graphics-based ITS STEAMER, where learning sessions are largely directed by the learner and the computer itself does not seem to do anything that could be called intelligent. But as Papert (1980, Ch. 7) and Hollan et al. (1984) stress, AI is in fact fundamental to the design and construction of both systems. First of all, the LOGO language itself is a variant of the AI programming language LISP; STEAMER's graphic editors and inspectible simulations are based on LISP and AI object-

oriented programming techniques. Neither would be possible without the power of these languages. Secondly, in both STEAMER and LOGO the interface through which the learner interacts with the computer draws heavily on the interactive programming environments developed for AI, and again neither would be possible without it. But most importantly the microworlds themselves derive from an analysis of how people reason about geometry or dynamic systems, of the knowledge that is to be learned, and of the way that knowledge has to be represented if learning is to be effective. Such essentially cognitive analyses, and the computational models that implement them, are AI's greatest achievements and promise. The resulting knowledge representations are built into the fabric of the microworlds—into the objects the learner can manipulate, the physical setup, her possibilities for action, the tasks she can perform, the tools she has to do them. Microworlds are often engaging because they are "realistic" and apparently relevant, as are simulations and "authentic" materials. They are also, like games and other informal learning environments, fun and self-motivating. But it is the cognitive analysis and the presentation of knowledge through limiting the possibilities explored by the learner (as contrasted with explanations or expository texts), not the motivational effects, that distinguish the AI-based applications from "traditional" approaches to education.

The educational philosophy lying behind the construction of Papert's mathematics and physics microworlds is one of "discovery" learning, of learning-by-doing, of giving the learner the opportunity and tools to learn by himself, rather than trying to teach him. While Papert's arguments are, in my view, compelling, simply letting a learner explore a microworld—or a complex simulation, or a "natural" environment in which learning might take place, for that matter—without any "guidance" whatever, has several limitations that should be noted:

- learners may form grossly incorrect models (conceptions) of the domain they are to "learn";
- learners do not know the cause of their errors, or even when they make them, so errors are non-constructive;
- learners may not explore all of interesting parts of the microworld, getting stuck in a small subworld;
- learners may not get into fine structure, even if they have mastered the grosser features;
- learners may not explore the microworld effectively, e.g. cycling in an incorrect procedure or fixating on complex problems before their simpler components have been encountered;
- learners may learn slowly, spending a lot of time on irrelevant activities;
- learners may not see the context of the simplifications in the microworld or the limitations of the specific tasks or setting.

These reservations argue for some form of instructional intervention or guidance, and even give some clues as to the most important issues in choosing what form that guidance should take. The key questions are empirical:

- how do people learn? and
- how can that learning be enhanced?

Considerable progress has been made for several domains, notably mathematics, physics, and engineering, where microworlds have been designed. I shall return to these questions when "increasingly complex microworlds" (Burton et al., 1984) and the Collins-Brown-Newman (1987) framework are discussed in section 4.2.

3. THE ARCHITECTURE OF ITS

Intelligent tutors as described in sections 2.1 and 2.2 are comprised of seven interdependent "architectural" components which perform the various functions necessary for teaching. The architecture can also serve as a convenient way of organizing discussions of the human tutors, coaches, and even classroom instructors that the ITSs are meant to emulate, since they, of course, must meet these functions as well. The components are:

- *domain knowledge*, i.e. expert knowledge of the subject matter to be taught, which the ITS uses to solve problems, generate explanations, etc.;
- *a student model*, i.e. a representation of the student's knowledge of the domain, often including a history of the student's responses;
- *diagnostic methods* for updating the student model as a function of the student's responses;
- *teaching knowledge*, i.e. knowledge of how to teach effectively in particular circumstances, along with a set of teaching tactics for accomplishing that;
- *planning and control mechanisms* which, on the basis of the current state of the student model and the domain and teaching knowledge, determine what to do;
- *an environment*, i.e. a set of tasks (activities) that the learner is to perform, and the tools given him to do so; and
- *an interface* by which the tutor and the student communicate.

ITSs differ in how these components are actually constructed, how they are interrelated, and what emphasis is given to each in the research and development process. Still, to build functioning prototypes on a computer, ITS designers and researchers have had to be very specific about how each is to be implemented. Although for expository purposes the components are treated as if they were separate modules, they must to some extent be designed and evaluated concurrently, since after all they must work together to produce effective learning. Whether or not it would be possible to replicate this work by building an ITS for foreign language instruction, it is my contention that considering how the issues have been addressed in ITS helps define the interesting research and instructional design issues for the field. In this section I discuss these components of ITS architecture and review some of the approaches ITS researchers have taken in each.

3.1 Domain Knowledge

As noted in section 2.2, in order for an ITS to provide flexible instruction and respond to arbitrary and unanticipated student responses, it must include an "expert system" which is able to solve problems in the domain that it is teaching. For the most ambitious foreign language tutor this would require a program that could understand natural language in very unstructured settings, a remote goal given the current state of the art in AI. For beginning and intermediate lessons (Feuerman et al., 1987; Xerox, 1985), or for learning sublanguages in structured technical domains (Geesey et al., 1989), however, the requisite domain expertise is currently within reach.

The expert knowledge components of existing ITS fall into three broad groups (Anderson, 1988). This first consists of "black box" experts, complex algorithms or simulations which can solve the problems but in which knowledge is not explicitly represented and the processes of solution are not useful for instruction. McArthur's algebra tutor (McArthur, 1986; McArthur et al., 1987), which uses the algebraic programming system REDUCE; the gaming tutors WUSOR (Goldstein, 1982) and WEST (Burton and Brown, 1982), which contain algorithms to compute the optimal move for any given position; and SOPHIE-I (Burton et al., 1982), which uses a general purpose circuit simulator, are examples. In language instruction most current parsers would fall into this class. At the other extreme are "glass box" experts (Goldstein and Papert, 1977), elaborate cognitive models and qualitative process models which represent the domain knowledge and reason about it in the same way that human beings do, so that describing and observing the expert's problem-solving process would be a useful component of the instruction. Anderson's LISP and Geometry tutors (Anderson and Reiser, 1985; Anderson, Boyle, and Yost, 1985) and SOPHIE-III (Brown and Burton, 1987) are examples. In foreign language detailed psycholinguistic models would be required here. In between are expert systems, developed using knowledge engineering techiques, which represent knowledge in an explicit way but may not reason about it the way human beings do. (While they may have the same knowledge base as cognitive models, they manipulate it differently.) A classic example is GUIDON (Clancey, 1986b, 1987), whose expert system consists of a knowledge base containing facts and relations about the domain, knowledge of diagnostic techniques, and a set of procedures (inference rules and interpretations) which use the knowledge base to solve problems.

While it is not crucial how domain problems are solved, there are, nevertheless, problems that have to be addressed in one way or another if the domain knowledge is to be useful for tutoring. In order to help students, tutors have to able to explain how the correct answer is derived and guide students along a path to it. Glass box experts already do this. The evolution of the GUIDON case method tutors from classical expert system to a more humanlike model (Clancey, 1986b) was motivated by just such considerations. Black box and classical expert systems in ITS have been made more "articulate" by augmenting them with devices like Burton and Brown's (1982) "issues recognizers," or Clancey's (1987) "t-rules," which compare student and expert performance (however generated) and base tutorial actions on differences ("issues") they are designed to detect. In this regard issue recognizers automate some aspects of CAI branching, although in order to do so effectively they may need more information than the match between correct and incorrect answers.

Besides articulateness, tutoring imposes other demands on the domain knowledge component. A good tutor needs some knowledge of how incorrect or partial answers are generated if it is to help students get past these blocks. Many tutors represent incorrect knowledge in the form of "false facts," "mal-rules," and "buggy" procedures, to deal with this problem. In foreign language the CALLE tutor (Feuerman et al., 1987; Xerox, 1985), which uses the LFG processor developed by Kaplan and Bresnan (1982), has this happy property of being able to account for ungrammatical as well as grammatical responses. The problem of knowledge representation is further compounded by the fact that in many domains there are many correct responses and many ways to reach a given correct response, each containing useful pedagogical information. A single algorithm or expert system may not be sufficient to capture this variation; multiple representations and views of the expert knowledge may be required (Olsson, 1986). Thus, although to a large extent domain knowledge and teaching knowledge are separated in ITS, there are practical restrictions on the modularity of the system.

3.2 The Student Model

The maintenance of a nontrivial student model, i.e. a fairly detailed representation of what the student "knows," is another important property that ITS shares with successful human tutors. Student modelling is a form of cognitive modelling as practiced in cognitive science generally, and a sibling of user modelling in computer science. Although all three share many techniques and strategies, unique issues arise because of the demands of learning and instruction. A student model is a sine qua non for individualized instruction, since it is the basis for what problems to present, what to teach, and how to adapt teaching strategies to experience with the student. At the least the student model requires a systematic description of the skills and subskills to be learned, but very often it is as elaborate as the "expert" model of the domain, containing not only the student's knowledge (correct and incorrect) but her goals and strategies for employing it. Since presumably the goal of learning is for the student to become more like an expert, the differences between the two are relevant to instruc-

tion, and the student and expert models cannot be incomparable. A great range of student models, differing in the kind of data required, the way knowledge is represented, and relationship to the expert model (VanLehn, 1988b), have been used in ITS.

The simplest student models are what Carr and Goldstein (1977) call "overlay" models, in which the student's knowledge is represented as the subset of expert skills which he has mastered. The most complex models, e.g. those maintaned by the LISP and Geometry tutors of Anderson and his colleagues (Anderson, Boyle, and Reiser, 1985; Anderson et al., 1988; Anderson, 1988), are elaborate computer simulations accounting in detail for the student's behavior. A important characteristic of most student models in ITS is the representation of both correct and incorrect knowledge, i.e. student models represent misinformation and "buggy" procedures as well as a simple lack of knowledge as in overlay models. Using the student model, an attempt is made to account for which specific errors are made and why, so that remediation can be tailored to the problems of the individual student. In both the decomposition of skills and the explicit representation of errors ITS models contrast with the parametric representation of the student—how much he knows or his level of mastery—commonly maintained in CAI and in the psychometric models underlying classical test theory and item response theory. Moreover, to the extent that student models are cognitive models, representing mental processes that lead to specific behaviors of particular individuals, student models are complex causal structures requiring more data for fitting than the answers given on tests or the final solutions to problems. I shall discuss the implications of these differences in a subsequent paper on research strategy for foreign language learning.

Two examples of ITS student models which would be realizable in a foreign language tutor and could be used in research to track student learning are the procedural networks used by Burton and Brown (Brown and Burton, 1978; Burton, 1982) in their WEST tutor; and Goldstein's (1982) genetic graphs, used by his WUSOR program to tutor strategies for the reasoning game Wumpus. In a procedural network the student's knowledge is modelled by analogy with a computer program, with a set of (LISP) procedures (or e.g. PASCAL subroutines) representing the subskills, connected by a control structure corresponding to calls in a computer program. Several different procedures which accomplish the same thing may be represented, as can "buggy" procedures which result in incorrect responses. (Control structures, however, do not contain bugs, nor are there problems, like limitations of working memory, which are built into more elaborate models.)

In a genetic graph the nodes correspond to procedural skills, rules, strategies, misinformation, and lack of information that the players may or may not have. The relationships between the nodes (links) are typed to represent evolutionary and logical relationships between them, such as refinement, generalization, specialization, analogy, and prerequisite; the typed links distinguish the genetic graph from a simple skill network. To add further structure, nodes (rules) can be grouped into "islands" to capture separability; and declarative facts justifying rules can be added. The student model is an overlay of the genetic graph, not an overlay of the final, perfect skills which the expert possesses. It thus contains errors and partial knowledge as well as the components of expert competence. Genetic graphs have obvious implications for tutoring, e.g. "tutor at the frontier of the student's position in the graph" and "use the links as a basis of explanation," although other tutorial principles (e.g. "vary examples") come into play as well.

Both procedural networks and genetic graphs contain a very fine breakdown of skills and subskills so that the tutor can focus on the specific problems the student has. This should certainly be possible for foreign language. Specification of the relationships between subskills, either in terms of calls or typed links, would be more difficult, but the rationale behind the sequencing of presentations in curricula might be a help. Evolutionary relationships, and the explicit representation of errors that occur at various stages, are particularly appealing for foreign language teaching because they could capture some of the implications of interlanguage studies and stage theories of language learning.

The most elaborate student models, those maintained by Anderson's LISP and Geometry tutors, are comprised of a set of rules, like the expert rules, which when fired simulate the student's behavior in detail. (The technique is called "model tracing.") Producing such models is very labor intensive, since currently they require hand coding malrules to capture all of the errors that are likely to be produced. Moreover fitting them for an individual student requires a substantial amount of very detailed data. While the interface of Anderson's tutors is constructed to ensure that the requisite data is available, this device may not be possible in domains such as foreign language learning, where responses follow quickly on one another without much intervening calculation. Still Anderson's success and the power of processors like LFG do not rule out model tracing as a viable long-term strategy.

3.3 Diagnostics

The diagnostic component of ITS updates the student model, inferring his knowledge from his responses, or to put it another way, "the student model is a data structure, and diagnosis is a process that manipulates it" (VanLehn 1988b). Diagnosis is clearly essential if instruction is to be adapted to the needs and problems presented by particular students as instruction progresses. Moreover, if teaching strategies are to be adaptive, assumptions about student behavior on which they

are based have to be tested. Thus diagnosis is intimately linked not only to the student model but to the planner as well. As Olsson (1986) stresses, the question is not so much "what is in the student's head?" as "what do we need to know in order to teach?" For complex activities, detailed cognitive models and a rich array of teaching tactics, we may need to know a good deal.

The diagnostic methods being developed in ITS may be its greatest payoff for the foreign language field, where work on assessment and testing has been dominated by considerations of how much a student knows, and where test results have been used for purposes of statistical comparison and certification, rather than for the important didactic goals of determining what a particular student knows at a particular point in time. Furthermore, continuing the contrast between diagnosis in ITS and ability testing, for teaching it is generally not sufficient to know whether or not a student has mastered a particular skill or gets a question right or wrong: it is equally important to know exactly what errors he makes and why he might have made them. The treatment of error indicates a third difference between diagnosis and ability testing, namely that in ITS diagnosis most errors are treated as systematic, not random, although allowance is made for performance lapses due to fatigue, boredom, memory failures, distractions, and the like.[4] Diagnosis, then, differs from testing in method as well as intent. I shall return to the research implications in the sequel.

VanLehn (1988b) distinguishes nine types of diagnostic techniques in ITS based on his typology of student models, but here it will be sufficient to discuss the three broad classes suggested by Olsson (1986). All support student models which at least contain a detailed specification of the skills and subskills being taught. The simplest diagnostic methods relate to overlay models, where the student's knowledge is described as a subset of expert knowledge, without any particular attention paid to misinformation or distortions (i.e. "errors"). Overlay methods are quite similar to those of CAI and ability testing, with the stipulations that the skills to be "tested" are highly disaggregated and mastery of each subskill evaluated. As a consequence, overlay methods are practical for use outside the context of ITS. For example Marshall (1980, 1981), not herself an ITS researcher but drawing on its methods, has developed several algorithms for choosing problems for presentation in an adaptive testing framework which enable overlay models to be fitted efficiently.

A second class of models focuses on error descriptions, attempting to account for specific errors at the behavioral level (e.g. specific incorrect answers to a problem or exercise) by postulating computational mechanisms that could produce them. Burton's analysis of "bugs" in multicolumn subtraction is an example (Burton, 1982; Brown and Burton, 1978). Bugs, i.e. incorrect procedures which may or may not produce errors depending on the problem, are represented in the procedural network (described in the previous subsection) along with correct procedures. Predictions of responses to problems presented are calculated for all possible correct and buggy procedures and the best-fitting model selected. For reasons discussed in Burton (1982) this is a very difficult task and the algorithm that implements it is computationally very intensive. Nevertheless, with enough computer time, Burton's methods are possible, and interesting findings have been obtained.

The third type of diagnostic method in ITS is simulation, of which Anderson's "model tracing" procedure is perhaps the best example. Because they are so detailed and require such specific data, Anderson's simulations are very closely tied to the specific subject matter the tutor is teaching, and his methods are difficult to extrapolate to other contexts. Olsson and Langley (1988) have, however, suggested a simulation method, implemented in the computer program DPF (Diagnostic Path Finder), which can be used in many ITS domains, like language learning, where the extensive data required by model tracing is unavailable, but where a cognitive model is entertained. Using task analysis (e.g. a procedural network), selective search, and machine learning techniques, DPF predicts both the specific behavioral path taken by the student and the strategy (rules) he used to get from each point on the path to the next. Although it is promising because of its psychological underpinnings, DPF has yet to be used in an ITS.

3.4 Teaching Knowledge

In addition to knowledge of the domain and knowledge of the student, ITSs contain explicit representations of their knowledge of teaching. As noted in section 2.2 the separation of teaching knowledge from domain knowledge is an important difference between ITS from CAI. Tutors, of course, stand or fall on the quality of their teaching knowledge. The specific tactics used, which are derived mostly from observations of expert practice, and which for the most part have not been tested empirically in ITS, are ITS's suppositions about how to successfully support learning of a given skill. By and large teaching knowledge in ITS is not that different from what one might obtain if perceptive CAI designers verbalized why they present the examples and explanations they do. The same might be said for teaching strategies, where a great deal is to be learned from the educational and training literature on instructional design. The explicitness and detail required by ITS, however, make it easier to test and modify, both in the design stage of the ITS and in a more experimentally oriented evaluation.

The content of the didactic component, i.e. the style of teaching used and how it is represented, varies considerably among ITSs, depending both on educational philosophy and on the kind of knowledge (declarative, procedural, qualitative causal models) they are trying to

communicate. The range includes:

- rules of thumb, governing case method dialogues, presentation of examples, and styles of explanation, as represented by the literally hundreds of t-rules in GUIDON (Clancey, 1987, Appendix E);
- case and example selection rules and teaching goals of Collins and Stevens's inquiry teaching and Socratic tutors (Collins and Stevens, 1982, 1983; Collins and Grignetti, 1975);
- principles of effective coaching, enunciated by Burton and Brown (1982) for the arithmetic game WEST (see section 4.2); and
- elaborately rationalized strategies, based on a theory of cognitive learning, employed in the LISP and Geometry tutors of Anderson and his colleagues (see section 4.1).

Some AI-based instructional systems, like STEAMER, have no real didactics at all, and because of this they are more closely allied to the computer-based microworlds discussed in section 2.3.

In an important review of ITS, Olsson (1986) distinguishes teaching tactics, the specific actions that tutors can take, from teaching strategies (to be discussed presently), which connect subject matter analysis and the current state of the student model with the tactics. Not surprisingly Olsson finds that in order to provide adaptive instruction, a tutor must have a wide range of instructional actions to choose from (his Principle of Versatile Output), but that unless conditions under which a particular tactic is to be evoked can be identified, the tactic will not increase the power of the system (his Principle of Strategic Repertoires). Considerable research is required to determine what those conditions are, in foreign language teaching or any other field.

3.5 Planning and Control Structures

As Olsson (1986) has emphasized, all ITSs need to be able to generate a teaching plan—or as he calls it a teaching strategy—on the basis of their current tutorial goal, their knowledge of the subject matter, and their assessment of the student. The plan is concerned, among other things, with the sequencing and selection of materials and with the form that the presentation should take. Moreover ITSs must be able to change their plans if they are not successful and if their assumptions about the student prove incorrect. Clearly this kind of planning must be based on general knowledge of pedagogy and assumptions about what is likely to work for a particular student at a particular point, and because of this, at a behavioral level at least, planning is not differentiated from teaching tactics in many ITS implementations. It is worth maintaining the distinction, however, because in the ITS literature planning points to different, and equally interesting, directions for empirical research than simply establishing what works.

The development of planning mechanisms for tutoring is a relatively neglected area in ITS, and tutors which tackle the issues head on are only now reaching the prototype stage. As Peachey and MaCalla (1986) point out, however, planning has a long history in AI in connection with determining the physical actions of real or simulated robots, and impressive results have been obtained. They also note that planning is more difficult in ITS than in robotics, because in the latter the plan is designed to change the state of the physical world, which can be observed; while in ITS the state that needs to be changed is student's knowledge, which is generally not fully observable. Furthermore students are intelligent, independent actors and in this regard, unlike the worlds of many robots, not entirely predictable. Recent work in ITS by Peachey and MaCalla (1986), Macmillan and Sleeman (1987), Russell (1987), and Murray (1988) promises to bring general planning methods to a practical stage.

Attention to results on planning in AI and ITS are important for research in foreign language learning because, as Macmillan and Sleeman have stressed, it is fundamental to the way human tutors look at their work. Recent empirical work by Leinhardt and Greeno (1986) and by Leinhardt and her colleagues at the Learning Reasearch and Development Center (Leinhardt, 1986; Leinhardt and Smith, 1985; and Leinhardt et al., 1987) has applied AI planning ideas to an analysis of the difference between novice and expert teachers. The increased richness of the characterization of teaching behavior which these concepts allow opens up many new avenues of research. Methods of relating behavior to plans—the obverse of planning, called "plan recognition" in AI—like cognitive modelling and diagnosis, are difficult, time consuming, and largely qualitative. Students too have intentions and plans which govern their interactions with teachers and computers; these also must be represented and established if student behavior is to be understood and, from a pedagogical point of view, if they are to be helped. The success of Johnson and Soloway's PROUST tutor (1987) in inferring the plans and intentions of beginning programmers in PASCAL from errors in their code, and the ability of Wilensky's and his students' Unix Consultant (Wilensky et al., 1984; Wilensky et al., 1986) to infer what the user really wants to know and do from often vague and ambiguous requests, similarly indicate interesting empirical research directions which will be discussed in the sequel.

Another relatively neglected area of concern in ITS is the whole question of curriculum, i.e. the selection and sequencing of topics for instruction. Because of its research and development orientation and the difficulty of encoding all of the knowledge required, ITS designers have been concerned for the most part with prototyping and producing instruction at the level of the lesson. Questions of how the knowledge to be mastered relates to other knowledge that would be presented in a course, and how the student's engagement with the ITS relates to the rest of his educational experience, have not arisen.

But as Lesgold (1988) has persuasively argued, curriculum development, like planning at the level of the lesson, requires a careful analysis of the goal structure of knowledge (prerequisites, dependencies, part-whole relations) as well as a domain expert which captures its cognitive content. To a certain extent the procedural networks and genetic graphs discussed in section 4.2 address these issues. A related approach is that of the "curriculum information networks" (CIN) used by the BIP tutors for the computer language BASIC (Westcourt et al., 1977; Westcourt et al., 1981). A CIN is a skill network with labeled links encoding such relationships as prerequisite, analogue, harder-than, component-of, kind-of, and functional dependency; unlike genetic graphs and procedural networks it does not represent misinformation. BIP takes the CIN, an overlay model of the student, and a catalogue of prewritten exercises and makes decisions concerning which exercise to present, thus automating many of the branching decisions built into the branching structure of CAI. As Halff (1988) has noted, the literature on instructional systems design is very useful for the task analysis which this kind of curriculum planning entails. Lesgold (1988), Bonar, Glaser, and their students at the Learning Research and Development Center at the University of Pittsburgh are designing a tutoring architecture, using object-oriented programming, focusing on knowledge goals, but this work is still in the development stage.

3.6 The Interface and the Environment

The last two components of ITS architecture, the environment and the interface, comprise the computer as experienced by the learner. The environment consists of the tasks the student is given and the tools he is given to perform them. It shades off into the interface, which determines how the student interacts with the tutor and with the domain. Many of the issues and alternatives in the design of ITS environments have already been discussed in the section on computer-based microworlds. Factors listed by Burton (1988), in an insightful review, cover many of the issues in ITS in general: the knowledge to be learned (communicated); the appropriate level of abstraction; the fidelity (verisimilitude), in various respects, with which the knowledge needs to be represented; sequencing of tasks, and adaptation of tools and props to the stage of learning; the amount of structure imposed on exploration of the environment by task definition; and the help provided (assistance in doing parts of the problem, aiding the learner to reflect on his own performance and skills, coaching, etc.).

Interface issues are closely related to teaching tactics and to those aspects of the implementation of the teaching plan having to do with dialogue structure, but different psychological and sociological considerations are involved. Although the interface has received relatively little systematic attention in the ITS literature (see, however, Hollan et al., 1984; Frye and Soloway, 1986; and Wenger, 1987), this is beginning to change as a result of increased interest in the general issues of interface design in computer science and AI. (See Miller, 1988, for a thorough review of the state of the art and its implications for ITS.) One such issue, for example, is effective on-line help, a common problem for most computer systems and applications. Only a fine line separates help from coaching and tutoring, since to achieve their immediate goals users often need to acquire some understanding of what they really want to do, how the system works, and what their options are. Some very interesting research, closely allied to ITS, is being carried out in the design of intelligent help systems (e.g., Wilensky et al., 1984; Wilensky et al., 1986; Fischer, 1988). Of particular note is the empirical and theoretical work of Breuker and his colleagues (Breuker, 1986; Winkels and Sandberg, 1987; Winkels et al., 1986) focusing on coaching and teaching strategies. Intelligent help may be feasible where full-blown intelligent tutoring is not because the limited nature of the domain makes knowledge representation possible, and because help requires less by way of explanation and actual problem solving than tutoring. User modelling, planning, and effective explanatory tactics are still necessary, however. With the advent of CD-ROM and other multimedia learning environments, interface issues will become very much more severe. Careful design will be required if these new technologies are to be effective.

There are important considerations having to do with learning which bear on the design of the interface. In Anderson's tutors the interface is carefully designed to hold some of the information about the problems being solved on screen so as to minimize burdens on short-term memory and allow students to concentrate on acquiring effective procedures (Lewis et al., 1987; Anderson, Boyle, and Reiser, 1985; Anderson et al., 1987). The interface also provides Anderson's model tracing procedures with the information needed to infer the learner's state of knowledge and problem-solving strategies. (In this respect it is the functional equivalent of think aloud and other verbal protocols used in cognitive science; see Ericsson and Simon, 1984.)

Since the interface in ITS (and CAI) determines the final form of the communication between the program and the user, the dialogue between the two could be managed at this level. Whether and to what extent the student or the ITS communicates in natural language then becomes an important issue, as yet to be resolved (see Burton et al., 1982, for discussion and experience with SOPHIE I, II, and III). Other interface issues which should be systematically explored—for CAI as well as ITS, and not only for computer-based instruction—include: the required speed of response of the tutor for various actions; screen management and the amount and type of information with which the user must deal; helping the student keep track of where he has been and where he is going during a session; and the pedagogical

use of computer graphics and other visual aids. The research agenda for ITS and CAI interfaces will be discussed in detail in a subsequent paper.

4. ITS AND LEARNING

From a research point of view the fundamental questions about foreign language learning are "what do people know?" and "how do they learn it?" Relatively concrete answers to these questions are also essential to good teaching and to the design of tools to support it. As pointed out above, AI researchers have had to think carefully about learning and effective teaching because they have had to build explicit rationales and prescriptions into their didactic knowledge bases, teaching strategies, and planning mechanisms, on the one hand, and into the design of computer-based microworlds and the environments and interfaces of ITS, on the other. Much of the empirical analysis of ITS and microworlds is directed toward testing these assumptions. The problems of applying learning theories developed in the substantive domains studied by ITS (e.g. mathematics, the maintenance and repair of complex equipment, and programming) to foreign language learning should certainly not be minimized. One of the hardest-learned lessons of ITS, and AI in general, is how domain-specific knowledge and successful procedures really are. Nevertheless it is encouraging that for some complex, nontrivial skills, theoretical models can be formulated which make it possible to study learning in some depth and to design successful learning environments in a principled way.

In this section I review two approaches to the conceptualization of learning which have guided the development of ITS and which derive from experiences with it. The first, exemplified by Anderson's ACT* theory of skill acquisition and the tutoring principles derived from it, brings a general theory of learning and cognition to bear on the design of tutors for very specific skills. The second, exemplified by the cognitive apprenticeship framework of Collins, Brown, and Newman, synthesizes a wide range of experiences with apprenticeship, microworlds, coaching, and ITS, in the form of a heuristic guide to the design of learning environments. While these theories are very different in form and motivation, each in its own way suggests interesting new avenues of research and development in foreign language learning.

4.1 Theories of Skill Acquisition

The deepest and most explicit formulation of learning principles motivating an ITS is a major theory of learning for cognitive skills in its own right, Anderson's ACT* theory of cognitive performance and skill acquisition. Indeed the design principles underlying the text processing, LISP, Geometry, and Algebra tutors built by Anderson and his colleagues at Carnegie-Mellon were derived from ACT* and are specifically intended as an empirical test of ACT*'s validity. As a cognitive theory ACT*'s concepts and propositions stand in stark contrast to the behaviorism that underlies most CAI. It is an elaborate, complex, evolving theory (the latest incarnation is called PUPS), and only the main points related to tutoring can be noted here. (For further discussion see Anderson, 1983, 1986, 1987, 1988; Anderson and Reiser, 1985; Anderson, Boyle, and Reiser, 1985; Anderson, Boyle, and Yost, 1985; Anderson et al., 1988; Anderson et al., 1987; and Lewis et al., 1987.)

The principal foci of ACT* are on the structure and operation of short-term memory and the nature of knowledge. Short-term memory is postulated to be very limited; many errors observed in cognitive tasks are accounted for by these limitations, not by lack of understanding. With regard to knowledge, following Winograd (1975) and a large literature in AI, a basic distinction is made between declarative knowledge, which is encoded quickly in schema-like structures, without reference to how it will be used; and procedural knowledge, which is embodied in highly efficient and use-specific forms and acquired through putting declarative knowledge into practice. Procedural knowledge takes the form of a production system (see also Klahr et al., 1987), i.e. a set of condition-action rules, with conditions including goals, such that if a condition is encoded in working memory the action should take place. In a formal sense production rules are like the rewrite rules of a grammar, but in an ACT* model of language acquisition (e.g. Anderson, 1987, Ch. 7) they would not be restricted to syntax. There are many hundreds of such rules, describing correct and incorrect ("buggy") programming in LISP and theorem proving in geometry, in the models of expert and student knowledge in Anderson's tutors. In a language tutor semantic, syntactic, pragmatic, and discourse rules, expressed in terms of what actions have to taken to achieve communicative goals, would serve this function.

At the beginning of a process of skill acquisition people work from declarative knowledge, holding it in working memory and transforming it into behavior in a calculated way by various "weak" methods such as analogy. As learning progresses more automatic and faster procedural knowledge is built up by a basic mechanism called "knowledge compilation." The term implies a strong analogy with the compilation of computer programs (see Wilensky, 1986), where a high-level code employing general and easy to use representations (e.g. FORTRAN) is translated into a lower-level (object) language that eliminates a lot of the overhead (blind search, testing conditions that do not arise in the task at hand, and inefficient deductions, etc.) of converting the high-level language into machine instructions (physical actions). As with computer languages, the cost of compilation is unintelligibility in the code, and this, along with the fact that procedural knowledge is very situation-specific, is what accounts for the difficulty in verbalization often associated with expert knowledge. Compila-

tion takes two main forms in Anderson's theory: "proceduralization," where specific rules for special cases are derived from general rules; and "composition," where several separate rules that might be applied sequentially are converted into a single procedure. ACT* also has a mechanism called "strengthening" which further speeds up performance. Inductive mechanisms, however, such as discrimination and generalization, common in other cognitive theories and in other production system models of learning (Klahr et al., 1987), play no role in ACT*.[5]

The teaching strategies of ACT* tutors are, then, derived from this general theory. ACT* tutors are very directive. They are designed to facilitate knowledge compilation by seeing that the student comes to the correct final procedural code. Their methodology is called "model tracing," which takes a correct model of the skill and the student's errorful procedures, finds a path to the correct model, and insists that the student stay on it. Since the end result of the tutoring process is knowledge-in-use (procedural knowledge), general instruction is given only in the context of solving specific problems. Goals, the conditions of the production rules, are made explicit. Since ACT* does not contain inductive mechanisms, carefully juxtaposed examples to guide induction are unimportant: students are simply told what the critical features are. Finally immediate feedback on errors and various aspects of the tutor's interface (the information displayed on the computer screen) helps students to manage working memory, so that they do not make mistakes due to memory overload and can concentrate on developing understanding and procedural skills.

Anderson's ACT* (PUPS) is not the only general theory of skill acquisition related to ITS. VanLehn and Brown's Repair Theory (VanLehn and Brown, 1980; Brown and VanLehn, 1982; VanLehn et al., 1984), and VanLehn's subsequent elaborations of the learning mechanisms in his Sierra learning simulation program (VanLehn 1985a, 1985b, 1988a, 1988b; Wenger, 1987, Ch. 8) represent equally comprehensive but quite different approaches to cognitive performance, learning, and knowledge representation (procedural skills are represented as generalized AND/OR graphs, as opposed to the production systems). Repair theory derives from attempts to explain data on the nature, origin, and remediation of "bugs" observed in Burton and Brown subtraction tutors (Brown and Burton, 1978; Burton, 1982; Burton and Brown, 1982). (Bugs are systematic but flawed procedures, which not only account for which problems will be answered incorrectly but precisely which wrong answers will be given—more than one hundred simple bugs, and many compound bugs have been described just for multicolumn subtraction!) The key concept is that of "impasse" or failure-driven learning, in which incomplete or "buggy" procedures lead to evident errors and intended actions that cannot be performed. Errors and impasses are overcome, often incorrectly, by ad hoc problem solving or "repairs," which produce new procedures on the way to mastering the skill. Besides the skill representation, the theory consists of a set of heuristic mechanisms that generate the possible repairs and a set of "critics" that filter out some repairs as unacceptable. While the intuitive underpinnings of Repair Theory and its success in accounting for systematic errors are very suggestive for language learning, the specific mechanisms discussed by Brown and VanLehn cannot be extrapolated as easily as ACT*'s from the domain (multicolumn subtraction) they are designed to explain. As VanLehn (1988a) points out, however, similar ideas are found in the work of Wexler and Culicover (1980) and Berwick (1985), so this general approach merits careful study by language researchers.

4.2 A Framework for Studying and Designing Successful Learning Environments

As attractive as they might be on general scientific grounds, the difficulty and expense of building large and comprehensive computer models like ACT* and Sierra make their implementation for foreign language learning at best a long-term goal. A product of ITS research which is of more immediate relevance to the study of foreign language learning is the framework constructed by Collins, Brown, and Newman (1987) to facilitate consideration of pedagogical and theoretical issues that arise in designing or evaluating learning environments. As a guide to design it specifies conditions for the successful construction of ITS, microworlds, and even classroom-based learning environments. As a guide to research it enumerates the main factors that that must be taken into account in empirical studies of what actually works.

The framework is based on a wide-ranging and insightful analysis of the common conditions associated with successful learning in a number of diverse settings:

- traditional apprenticeship for occupational skills such as tailoring (Lave, in preparation);
- a detailed analysis of three "success models" in schools—Palinscar and Brown's (1984) reciprocal teaching method for reading; Scardamalia and Bereiter's (1983, 1985) procedural facilitation for writing; and Schoenfeld's (1983, 1985) methods of teaching mathematical problem solving—which are characterized by the same principles as apprenticeship and accordingly termed "cognitive apprenticeship";
- skiing instruction, a success model for a complex procedural skill (Burton et al., 1984);
- microworlds, as described in section 2.3 above, and reactive learning environments, e.g. SOPHIE; and, of course,
- intelligent tutoring systems.

It builds on the earlier paradigm of "increasingly complex microworlds" (ICM) of Burton et al. (1984; also Fischer, 1988), and is now being extended in the the work of Brown et al. (1988); see also Collins and Brown (1988).

The key concept underlying all of the success models is "situated learning," i.e. knowledge acquired in the social and functional context of its use. This is contrasted with formal schooling, which has tended to emphasize the orderly presentation of factual knowledge, abstracted (dissociated) from the context of its use. The framework consists of a checklist of factors that must be considered in the design of successful situated learning, and a set of suggestions, synthesized from the "success models" above, as to how to employ them. It has four major divisions: content, methods, sequencing, and sociology.

1. Content refers to the types of knowledge which are acquired in learning. Its main categories are factual and procedural (domain) knowledge on the one hand and strategic knowledge (heuristic strategies for accomplishing tasks; control strategies, such as planning at various levels, monitoring, and diagnosis; and learning strategies) on the other.

2. Methods refer to the kinds of help given the learner in acquiring skills and discovering knowledge. They include: making the sources and consequences of errors apparent; "modelling," to enable student to build a model of expert practice by making explicit what is largely implicit; coaching, offering help to bring the student closer to expert practice, and coaching aimed at executable advice (advice that can be followed; "scaffolding," providing supports for carrying out simplified but real tasks, and "fading," removing supports as skill develops; reflective comparison with experts, other students, and ultimately with the learner's inner cognitive model; and exploration of interesting subtasks and methods subsumed under a general goal.

3. Sequencing addresses changing learning needs in different phases of the acquisition process. The main considerations here are the management of increasing complexity and increasing diversity in the tasks the student is given; and coordinating instruction of local versus global skills. The elements of sequencing are addressed in detail in the earlier ICM framework. They include maintaining motivation through success, avoiding the dangers of oversimplification (unjustified extrapolation and an unwillingness to try new things), structuring the environment so that progressively less simple but still realistic versions of the target expert skill are learned, and using task specification to focus attention on important factors in the microworld.

4. Sociology refers to the social context of learning. Besides the methods of situating knowledge so that students come to understand its purposes and conditions of use, the category directs attention to exposure to examples of expert practice and active communication about expertise; intrinsic motivation in the tasks; and exploitation of cooperation and competition in the social situation of learning. It is here that the framework has the greatest bearing on classroom-based learning.

While the framework directs us to look in particular directions to characterize good and bad instruction—and effective and ineffective learning—much work needs to be done in variable specification and measurement in the foreign language field before the framework can be used in rigorous research. As with ACT*, experience with learning environments designed specifically according to these principles will clearly play an important role. It would be an interesting exercise for language pedagogues to reformulate the principles of successful language learning in these, if for no other reason than to clarify the similarities and differences between language learning and the learning of other cognitive and procedural skills.

5. CONCLUSION

By way of conclusion, I would like to note and comment on some of the main themes that emerge in the paper that will be developed in detail with regard to foreign language learning in the two following papers. First and foremost the focus of all of the work that has been reviewed here has been on learning. Designers of ITSs and microworlds have looked very concretely at what is to be learned, how it is to be learned, and how learning can be effectively supported. The artifacts that they have designed have been explicitly motivated by this analysis. By contrast many applications of advanced technologies to foreign languages have taken learning issues for granted, implicitly relying on accepted educational methodologies and other components of the larger system within which the new technologies are to be embedded to achieve their instructional goals. It is in the analysis of learning that the main interest of ITS for the foreign language field lies.

Second, the orientation of AI research in ITS and microworlds is essentially cognitive, although motivation is not ignored. The primary concern is with the acquisition of knowledge, and with the cognitive and metacognitive processes, and complex procedural skills, that put it to use. Models of expert and student knowledge are at the heart of ITS. Further, as Papert has so cogently argued, the design of tasks and microworlds is basically an epistemological enterprise, even though there is no specific tutorial expertise involved. Work in foreign language learning has put relatively little emphasis on the precise specification of cognitive mechanisms involved in learning as compared with other factors (e.g. presumed individual differences in motivation, learning, and cognitive styles). Redressing this imbalance, I shall argue, is essential for the design of effective foreign language learning.

Third, as in many other problems in AI, the representation of knowledge turns out to be the key research and design problem. Many would claim that it is in the area of knowledge representation that AI has had its greatest successes. Knowledge is represented in ITS not only in the data structures and manipulations of the expert model, but in the physical structures of its tasks,

interfaces, and microworlds. Both domain knowledge and goal structure need to be represented. If ITSs are to be built for foreign language learning it is the problems of knowledge representation that must first be tackled. The knowledge of language that can be usefully represented will determine the possibilities for ITS in foreign language learning, and this must be carefully assessed.

Fourth, while one can usefully talk about the elements of ITS architecture separately, it is striking how interdependent they are and, accordingly, how important it is to design them concurrently. Teaching must be adapted to the nature of the knowledge to be learned, and correlatively knowledge representation must take into account the demands of teaching. Diagnosis updates student models and at the same time guides the instructional planner: it is concerned with what we need to know in order to teach. The interface and the environment—the medium in which communication, interaction, and learning takes place—not only contain information about the domain, they also shape the way the learner reflects on her knowledge and on her own learning processes. They must accordingly be designed with learning considerations in mind.

Fifth, although I do not discuss this at all in the paper, it would be an interesting exercise to stand many of the criticisms of CAI and ITS on their heads and look at learning in the classroom in the light of the kinds of considerations that the builders of ITSs have faced. The first question would be the obvious one: given the knowledge to be communicated, and given what is known about the way people learn, would classrooms (a teacher, several students, etc.) be invented in the first place, and if so how would they be structured? Or to put it another way, for communicating knowledge, what is the comparative advantage of the educational technology presently in use, and for what kinds of knowledge is it particularly effective? With regard to learning environments and diagnosis, one could ask similar questions about materials and testing. (Of course there might be other reasons to invent classrooms, textbooks, and testing besides their efficacy in communicating knowledge.) I realize it is hard to treat such questions as anything but rhetorical or polemical, but they do have a core of scientific content which debates on education in our rapidly changing society cannot ignore.

Sixth, with regard to empirical research strategies, the sections on student modelling and diagnosis describes rigorous computational models and methods of analyzing data that point in very different directions from the statistical models and methods which have dominated the educational and evaluational literature on learning in instructional settings. The difference lies not so much in the formalisms employed, although these are very different indeed, but in the kinds of questions that are addressed and the goals that are served. Pace: of course empirical work on foreign language learning needs both. The second paper in this series will be devoted to a dicussion of these issues.

Finally, I would not want to exaggerate the extent to which any existing ITSs or microworlds have reached the educational goals that they have set. That is a difficult empirical question on which there has been lamentably little systematic research. As I indicated at the outset, however, most ITSs are prototypes whose primary purpose is research, and it is the results of this research that I have stressed. Here, the myriad ideas and methods developed have many fruitful and practical applications to foreign language learning, in ways that will be the subject of the third paper in this series.

NOTES

1. One might with some justification say that the major achievements of AI as a discipline have been primarily in the area of knowledge representation. On the declarative versus procedural representations see Winograd, 1975. With reference to ITS, see Clancey's (1986b, 1987) discussion of the motivation for NEOMYCIN as a basis for GUIDON; and Anderson (1988). Although a useful one, the declarative/procedural distinction cannot be pushed too far, as VanLehn (1988b) has cogently argued.

2. To anticipate a little, using insights generated by ITS research, misleading dichotomies, like learning versus acquisition, can be reformulated so that both have a role.

3. The word is used in many different senses, even by the same author (see Lawler, 1987); some key phrases, overlapping Pea's, are "limited, simplified slices of reality," "worlds with limited possibilities," "fixed and limited objects, properties, and relations," "problem spaces," and "task domains along with the tools to operate in them."

4. How systematic errors really are is, of course, an empirical question. Whether errors are treated as random depends on how important and how hard it is to characterize them, as well as the social and psychological aspects of assessment method in the domain.

5. Of course, conditioning, reinforcement, choice probabilities, and the rest of the conceptual repertoire of behavioristic theories of learning are irrelevant altogether.

REFERENCES

Abelson, H., and diSessa, A.A. 1980. Turtle Geometry: The Computer as a Medium for Exploring Mathematics. Cambridge, MA: MIT.

Anderson, J.R. 1983. The Architecture of Cognition. Cambridge, MA: Harvard.

———. 1986. Knowledge compilation: The general learning mechanism. In R. Michalski, J. Carbonnell, and T. Mitchell (Eds.), Machine Learning, v.2. Palo Alto, CA: Tioga.

———. 1987. Skill acquisition: Compilation of weak-method problem solutions. Psychological Review 94, 192-210.

———. 1988. The expert module. In Polson and Richardson (1988).

Anderson, J.R., and Reiser, B.J. 1985. The LISP tutor. Byte 10, 159-175.

Anderson, J.R.; Boyle, C.F.; and Reiser, B.J. 1985. Intelligent tutoring systems. Science 228, 456-458.

Anderson, J.R.; Boyle, C.F.; and Yost, G. 1985. The GEOMETRY tutor. In A. Joshi (Ed.), Proceedings of the Ninth International Joint Conference on Artificial Intelligence. Los Altos, CA: Kaufmann.

Anderson, J.R.; Boyle, C.F.; Corbett, A.; and Lewis, M. 1988. Cognitive modelling and intelligent tutoring. Draft. Pittsburgh: Carnegie-Mellon.

Anderson, J.R.; Boyle, C.F.; Farrell, R.; and Reiser, B.J. 1987. Cognitive principles in the design of computer tutors. In P. Morris (Ed.), Modelling Cognition. New York: Wiley.

Barchan, J. 1987. Language Independent Grammatical Error Reporter. Research Report, Computer Science Department, University of Exeter (UK).

Barchan, J., and Wusterman, J. 1988. A Prolog-based tool for grammatical analysis of Western European Languages. Research Report, Computer Science Department, University of Exeter (UK).

Barchan, J.; Woodmansee, B.; and Yazdani, M. 1986. A prolog-based tool for French grammatical analysis. Instructional Science 15, 21-48.

Berwick, R. 1985. The Acquisition of Syntactic Knowledge. Cambridge, MA: MIT.

Breuker, J. 1986. Coaching in help systems. In J. Self (Ed.), Intelligent Computer-Aided Instruction. London: Chapman and Hall (in press).

Brown, J.S., and Burton, R.R. 1978. Diagnostic models for procedural bugs in basic mathematical skills. Cognitive Science 2, 155-192.

Brown, J.S., and Burton, R.R. 1987. Reactive learning environments for teaching electronic troubleshooting. Advances in Man-Machine Systems 3, 65-98.

Brown, J.S., and Greeno, J. (Chairmen) 1984. Report of the research briefing panel on information technology in precollege education. In Research Briefings 1984. Washington, DC: National Academy Press.

Brown, J.S., and VanLehn, K. 1982. Repair theory: a generative theory of bugs in procedural skills. Cognitive Science 4, 379-426.

Brown, J.S.; Collins, A.; and Duguid, P. 1988. Cognitive apprenticeship, situated cognition and social interaction. Report no. 8. Palo Alto, CA: Institute for Research on Learning.

Brown, J.S.; Moran, T.P.; and Williams, M.D. 1982. The semantics of procedures: A cognitive basis for maintenance training competency. Palo Alto, CA: Xerox Corporation CIS Working Paper.

Burton, R.R. 1982. Diagnosing bugs in a simple procedural skill. In Sleeman and Brown (1982).

Burton, R.R. 1988. The environment module of intelligent tutoring systems. In Polson and Richardson (1988).

Burton, R.R. and Brown, J.S. 1982. An investigation of computer coaching for informal learning activities. In Sleeman and Brown (1982).

Burton, R.R.; Brown, J.S.; and Fischer, G. 1984. Skiing as a model of instruction. In B. Rogoff and J. Lave, Everyday Cognition. Cambridge, MA: Harvard.

Burton, R.R.; Brown, J.S.; and de Kleer, J. 1982. Pedagogical, natural language and knowledge engineering techniques in SOPHIE I, II, and III. In Sleeman and Brown (1982).

Carbonell, J.R. 1970. AI in CAI: an artificial intelligence approach to computer-assisted instruction. IEEE Transactions in Man-Machine Systems 11, 19-202.

Carr, B., and Goldstein, I.P. 1977. Overlays: A theory of modeling for computer-aided instruction. Artificial Intelligence Memo 406. Cambridge, MA: MIT.
Cerri, S., and Breuker, J. 1981. A rather intelligent language teacher. Studies in Language Learning 3, 182-192.
Clancey, W.J. 1982. Tutoring rules for generating case method dialog. In Sleeman and Brown (1982).
———. 1984. Methodology for building an intelligent tutoring system. In W. Kintsch, J. Miller, and P. Polson (Eds.), Models and Tactics in Cognitive Science. Hillsdale, NJ: Erlbaum.
———. 1986a. Qualitative student models. In Annual Review of Computer Science. Palo Alto, CA: Annual Reviews.
———. 1986b. From GUIDON to NEOMYCIN and HERACLES in twenty short lessons. ONR Final Report (1979-1985). AI Magazine 7, 40-60.
———. 1987. Knowledge-based Tutoring: The GUIDON Program. Cambridge, MA: MIT.
———. 1988. The knowledge engineer as student: Metacognitive bases for asking good questions. In Mandl and Lesgold (1988).
Collins, A., and Brown, J.S. 1988. The computer as a tool for learning through reflection. In Mandl and Lesgold (1988).
Collins, A., and Grignetti, M. 1975. Intelligent CAI. BBN Report 3181. Cambridge: Bolt Beranek and Newman.
Collins, A., and Stevens, A.L. 1982. Goals and strategies of inquiry teachers. In R. Glaser (Ed.), Advances in Instructional Psychology. Hillsdale, NJ: Erlbaum.
Collins, A., and Stevens, A.L. 1983. Cognitive theory of interactive teaching. In C.M. Reigeluth (Ed.), Instructional Design Theories and Models: An Overview. Hillsdale, NJ: Erlbaum.
Collins, A.; Brown, J.S.; and Newman, S.E. 1987. Cognitive apprenticeship: Teaching the craft of reading, writing, and mathematics. In L.B. Resnick (Ed.), Cognition and Instruction: Issues and Agendas. Hillsdale, NJ: Erlbaum.
Dede, C.J.; Zodhiates, P.P.; and Thompson, C.L. 1985. Intelligent computer-assisted instruction: A review and assessment of ICAI research and its potential for education. Cambridge, MA: Educational Technology Center, Harvard University.
Ericsson, K.A., and Simon, H. A. 1984. Protocol Analysis: Verbal Reports as Data. Cambridge, MA: MIT Press.
Feuerman, K.; Marshall, C.; Newman, D.; and Rypa, M. 1987. The CALLE Project. Technical Report. Pasadena: Xerox Corporation.
Fischer, G. 1988. Enhancing incremental learning processes with knowledge-based systems. In Mandl and Lesgold (1988).
Frye, D., and Soloway, E. 1986. Interface design: A neglected issue in educational software. New Haven: Department of Computer Science, Yale University.
Geesey, R.; Ginsberg, R.; Lancaster, J.; Manukian, E.; and Reeker, L. 1989. Learning environments for scientific and technical competency. Unpublished manuscript.
Goldstein, I.P. 1982. The genetic graph: A representation for the evolution of procedural knowledge. In Sleeman and Brown (1982).
Goldstein, I.P., and Papert, S. 1977. Artificial intelligence, language, and the study of knowledge. Cognitive Science 1, 1-21.
Halff, H.M. 1988. Curriculum and instruction in automated tutors. In Polson and Richardson (1988).
Haugeland, J. 1984. First among equals. In W. Kintsch, J. Miller, and P. Polson (Eds.), Models and Tactics in Cognitive Science. Hillsdale, NJ: Erlbaum.
Hollan, J.D.; Hutchins, E.L.; and Weitzman, L.M. 1984. STEAMER: An interactive inspectable simulation-based training system. AI Magazine 5(2), 15-28.
Institute for Research on Learning. 1988. The Advancement of Learning. Palo Alto, CA: IRL.
Johnson, W.B. 1988. Pragmatic considerations in research, development, and implementation of intelligent tutoring systems. In Polson and Richardson (1988).
Johnson, W.L., and Soloway, E. 1987. PROUST: An automatic debugger for Pascal programs. In Kearsley (1987).
Kaplan, R.M., and Bresnan, J. 1982. Lexical-functional grammar: A formal system for grammatical representation. In J. Bresnan (Ed.), The Mental Representation of Grammatical Relations. Cambridge, MA: MIT.
Kearsley, G.P. (Ed.) 1987. Artificial Intelligence and Instruction: Applications and Methods. Reading, MA: Addison-Wesley.
Klahr, D.; Langley, P.; and Neches, R. (Eds.). 1987. Production System Models of Learning and Development. Cambridge, MA: MIT.
Lave, J. In preparation. Tailored learning: Apprenticeship and everyday practice among craftsmen in West Africa. Stanford: Institute for Research on Learning.
Lawler, R.W. 1987. Learning environments: Now, then, and someday. In Lawler and Yazdani (1987).
Lawler, R. W., and Yazdani, M. (Eds.) 1987. Artificial Intelligence in Education, vol. 1: Learning Environments and Tutoring Systems. Norwood, NJ: Ablex.

Leinhardt, G. 1986. Math lessons: A contrast of novice and expert competence. Journal of Research in Mathematics Education (in press).
Leinhardt, G., and Greeno, J.G. 1986. The cognitive skill of teaching. Journal of Educational Psychology 78, 75-95.
Leinhardt, G., and Smith, D.A. 1985. Expertise in mathematics instruction: Subject matter knowledge. Journal of Educational Psychology 77, 247-271.
Leinhardt, G.; Weidman, C.; and Hammond, K.M. 1987. Introduction and integration of classroom routines by expert teachers. Curriculum Inquiry 17, 135-176.
Lesgold, A. 1988. Toward a theory of curriculum for use in designing intelligent instructional systems. In Mandl and Lesgold (1988).
Lewis, M.W.; Milson, R.; and Anderson, J.R. 1987. The TEACHER'S APPRENTICE: Designing an intelligent authoring system for high school mathematics. In Kearsley (1987).
Macmillan, S.A., and Sleeman, D.H. 1987. An architecture for a self-improving instructional planner for intelligent tutoring systems. Computational Intelligence 3, 17-27.
Mandl, H., and Lesgold, A. (Eds.) 1988. Learning Issues for Intelligent Tutoring Systems. NewYork: Springer-Verlag.
Marshall, S.P. 1980. Procedural networks and production systems in adaptive diagnosis. Instructional Science 9, 129-143.
———. 1981. Sequential item selection: Optimal and heuristic policies. Journal of Mathematical Psychology 23, 134-152.
McArthur, D. 1986. Developing computer tools to support performing and learning complex cognitive skills. In K. Pedzek, D. Berger, and B. Bankes (Eds.), Applications of Cognitive Psychology: Computing and Education. Hillsdale, NJ: Erlbaum.
McArthur, D.; Stasz, C.; and Hotta, J.Y. 1987. Learning problem-solving skills in algebra. Santa Monica, CA: Rand Corporation Note.
Miller, J.R. 1988. The role of human-computer interaction in intelligent tutoring systems. In Polson and Richardson (1988).
Morgenstern, D. 1986. The Athena language project. Hispania 69, 740-745.
Murray, J.H.; Morgenstern, D.; and Furstenberg, G. 1987. The Athena language learning project: Design issues for the next generation of language learning tools. Draft. Cambridge, MA: MIT.
Murray, W.R. 1988. Personal communication.
Olsson, S. 1986. Some principles of intelligent tutoring. Instructional Science 14, 293-326.
Olsson, S., and Langley, P. 1988. Psychological evaluation of path hypotheses in cognitive diagnosis. In Mandl and Lesgold (1988).
O'Shea, T., and Bornat, R. 1987. A five component model for computer based training. Unpublished manuscript.
O'Shea, T., and Self, J.A. 1983. Learning and Teaching with Computers. Englewood Cliffs, NJ: Prentice-Hall.
O'Shea, T.; Bornat, R.; du Boulay, B.; Eisenstadt, M.; and Page, I. 1984. Tools for creating intelligent computer tutors. In A. Elithorn and R Banerji (Eds.), Artificial and Human Intelligence. Amsterdam: North Holland.
Palinscar, A.S., and Brown, A.L. 1984. Reciprocal teaching of comprehension-fostering and monitoring activities. Cognition and Instruction 1, 117-175.
Papert, S. 1980. Mindstorms: Children, Computers, and Powerful Ideas. New York: Basic Books.
———. 1986. Rethinking mathematics learnability in a computer culture. Unpublished lecture. Stanford University.
Park, O-C.; Perez, R.S.; and Seidel, R.J. 1987. Intelligent CAI: Old wine in new bottles, or a new vintage? In Kearsley (1987).
Pea, R.D. 1987. Integrating human and computer intelligence. In Pea and Sheingold (1987).
Pea, R.D., and Sheingold, K. 1987. Mirrors of Minds: Patterns of Experience in Educational Computing. Norwood, NJ: Ablex.
Pea, R.D., and Soloway, E. 1987. Mechanisms for facilitating a vital and dynamic education system: Fundamental roles for education science and technology. Final Report for Office of Technology Assessment, U.S. Congress.
Peachey, D.R., and McCalla, G.I. 1986. Using planning techniques in intelligent tutoring systems. International Journal of Man-Machine Studies 24, 77-98.
Polson, M.C., and Richardson, J.J. (Eds.) 1988. Foundations of Intelligent Tutoring Systems. Hillsdale, NJ: Erlbaum.
Psotka, J.; Massey, L.D.; and Mutter, S.A. (Eds.) 1988. Intelligent Tutoring Systems: Lessons Learned. Hillsdale, NJ: Erlbaum.
Russell, D.M. 1987. The instructional design environment: Interpreter. In Psotka et al. (1988).
Scardamalia, M., and Bereiter, C. 1983. Child as co-investigator: Helping children gain insight into their own mental processes. In S. Prais, G. Olson, and H. Stevenson (Eds.), Learning and Motivation in the Classroom. Hillsdale, NJ: Erlbaum.
Scardamalia, M., and Bereiter, C. 1985. Fostering the development of self-regulation in children's knowledge processing. In S. Chipman, J.W. Segal, and R. Glaser (Eds.), Teaching and Learning Skills: Research and Open Questions. Hillsdale, NJ: Erlbaum.

Schank, R.C. 1984. The Cognitive Computer. Reading, MA: Addison-Wesley.
Schoenfeld, A.H. 1983. Problem solving in the mathematics curriculum: A report, recommendations and an annotated bibliography. MAA Notes no. 1. The Mathematical Association of America.
Schoenfeld, A.L. 1985. Mathematical Problem Solving. New York: Academic Press.
Sleeman, D., and Brown, J.S. (Eds.) 1982. Intelligent Tutoring Systems. London: Academic Press.
Suppes, P. (Ed.) 1981. University-Level Computer-Assisted Instruction at Stanford: 1968-1980. Palo Alto, CA: Institute for Mathematical Studies in the Social Sciences, Stanford University.
Uren, J., and Yazdani, M. 1988. Spanish LINGER. Research Report, Computer Science Department, University of Exeter (UK).
VanLehn, K. 1985a. Acquiring procedural skills from lesson sequences. Technical Report ISL-9. Palo Alto, CA: Xerox Corporation.
———. 1985b. Learning one subprocedure per lesson. Technical Report ISL-10. Palo Alto, CA: Xerox Corporation.
———. 1988a. Toward a theory of impasse-driven learning. In Mandl and Lesgold (1988).
———. 1988b. Student modeling. In Polson and Richardson (1988).
VanLehn, K., and Brown, J.S. 1980. Planning nets: A representation for formalizing analogies and semantic models for procedural skills. In R.E. Snow, P.A. Federico, and W.E. Montague (Eds.), Aptitude Learning and Instruction, vol. 2: Cognitive Process Analysis of Learning and Problem Solving. Hillsdale, NJ: Erlbaum.
VanLehn, K.; Brown, J.S.; and Greeno, J.G. 1984. Competitive argumentation in computational theories of cognition. In W. Kintsch, J.Miller, and P. Polson (Eds.), Models and Tactics in Cognitive Science. Hillsdale, NJ: Erlbaum.
Wenger, E. 1987. Artificial Intelligence and Tutoring Systems: Computational and Cognitive Approaches to the Communication of Knowledge. Los Altos, CA: Kaufmann.
Westcourt, K.,; Beard, M.; and Barr, A. 1981. Curriculum information networks for CAI: Research on testing and evaluation by simulation. In Suppes (1981).
Westcourt, K.; Beard, M.; and Gould, L. 1977. Knowledge-based adaptive curriculum sequencing for CAI: Application of a network representation. Proceedings of the National ACM Conference, Seattle, 234-240. New York: ACM.
Wexler, K., and Culicover, P. 1980. Formal Principles of Language Acquisition. Cambridge, MA: MIT.
Wilensky, R. 1986. Common LISPcraft. New York: Norton.
Wilensky, R.; Arens, Y.; and Chin, D. 1984. Talking to UNIX in English: An overview of UC. Communications of the ACM 27, 574-593.
Wilensky, R. et al. 1986. UC—a progress report. Report no. UCB/CSD 87/303. Berkeley: Computer Science Division, University of California.
Winkels, R., and Sandberg, J. 1987. The EUROHELP coach. Memo 94 of the VF Project, Department of Social Science Informatics, University of Amsterdam.
Winkels, R.; Sandberg, J.; and Breuker, J. 1986. Coaching strategies and tactics for IHSs. Memo 78 of the VF Project. Department of Social Science Informatics, University of Amsterdam.
Winograd, T. 1975. Frame representations and the declarative/procedural controversy. In D.G. Bobrow and A.M. Collins (Eds.), Representation and Understanding: Studies in Cognitive Science. New York: Academic Press.
Winograd, T., and Flores, F. 1985. Understanding Computers and Cognition: A New Foundation for Design. Norwood, NJ: Ablex.
Xerox 1985. CALLE Project Final Report. Pasadena, CA: Xerox Special Information Systems, Vista Laboratory.
Yazdani, M. (Ed.) 1984. New Horizons in Educational Computing. New York: Wiley.
———. 1986. Intelligent tutoring systems: An overview. Expert Systems 3, 154-162.
———. 1988. Language tutoring with Prolog. Draft, Department of Computer Science, University of Exeter (UK).

About the National Foreign Language Center

The National Foreign Language Center (NFLC) is dedicated to improving the foreign language competency of Americans. NFLC emphasizes the formulation of public policy to make our language teaching systems responsive to national needs in international competitiveness and foreign relations. Its primary tools in carrying out this objective are:

•**Surveys**—NFLC conducts surveys to collect previously unavailable information on issues concerning national strength and productivity in foreign language instruction, and our foreign language needs in the service of the economic, diplomatic, and security interests of the nation.

•**National Policy Planning Groups**—In order to address major foreign language policy issues, NFLC convenes national planning groups that bring together users of foreign language services and representatives of the language instructional delivery systems in formal education, the government, and the for-profit sector.

•**Research**—NFLC conducts research on innovative, primarily individual-oriented strategies of language learning to meet the nation's foreign language needs of the future.

The results of these surveys, discussions, and research are made available through NFLC's publications, such as these *Occasional Papers*, and they form the basis of fresh policy recommendations addressed to national leaders and decision-makers.

NFLC was established in 1987 as part of The Johns Hopkins University with funding from four private foundations: The Exxon Education Foundation, the Ford Foundation, the Andrew W. Mellon Foundation and the Pew Memorial Trusts.